Ansible uses modules to complete the task on the remote server. The most commonly used command modules are “command”, “shell”, and “raw”. Each module has its own advantages and disadvantages. In Ad-hoc command mode, unless you specify the module name, it uses “**command**” module by default. In this article, we will see the functionality of these modules and use cases.

* [command – Executes a command on a remote node](https://docs.ansible.com/ansible/latest/modules/command_module.html#command-module)
* [shell – Execute commands in nodes.](https://docs.ansible.com/ansible/latest/modules/shell_module.html#shell-module)
* [raw – Executes a low-down and dirty SSH command](https://docs.ansible.com/ansible/latest/modules/raw_module.html#raw-module)

**Command Module:**

“Command” module is the default module in Ansible Ad-hoc mode. command module can able to execute only the binaries on remote hosts. Command module won’t be impacted by local shell variables since it bypasses the shell. At the same time, it may not be able to run “shell” inbuilt functions(ex: set) and redirection (which also shell’s inbuilt functionality).

1.Login to ansible server.

2. Let’s find out the current run level on host “uaans”.

[sysadmin@ansible-server ~]$ ansible uaans -a "who -r"

uaans | SUCCESS | rc=0 >>

run-level 5 2018-06-29 06:14

[sysadmin@ansible-server ~]$

You can also make the output in a single line using the “-o” option.

[sysadmin@ansible-server ~]$ ansible uaans -a "who -r" -o

uaans | SUCCESS | rc=0 | (stdout) run-level 5 2018-06-29 06:14

[sysadmin@ansible-server ~]$

The above command executed the below-listed binary in the remote server and got the desired results.

$ ls -lrt /bin/who

-rwxr-xr-x. 1 root root 49872 Apr 10 21:35 /bin/who

Same way, you should be able to execute the binaries on the remote host. To use “command” module, the remote host should have python installed (2.7 >).

3. Let’s look at the example if the remote server doesn’t have python installed.

[sysadmin@ansible-server ~]$ ansible uaans69 -a "who -r"

uaans69 | FAILED! => {

"changed": false,

"module\_stderr": "Shared connection to uaans69 closed.\r\n",

"module\_stdout": "/bin/sh: /usr/bin/python: No such file or directory\r\n",

"msg": "MODULE FAILURE",

"rc": 0

}

[sysadmin@ansible-server ~]$

$ ansible nodes -a "ls"

$ ansible nodes -a "touch f1 f2 f3"

$ ansible nodes -a "date"

“Command” module failed because the remote host doesn’t have python installed.

### ****Raw Module:****

Datacenter has combinations of appliances, servers and network devices. ***Python*** might not be installed on all the server and appliances. If that’s the case, how to manage those using Ansible?  Ansible offers “raw” module to overcome such a limitation of the “command” module.

[sysadmin@ansible-server ~]$ ansible uaans69 -m raw -a "who -r"

uaans69 | SUCCESS | rc=0 >>

run-level 3 2018-05-14 13:19

Shared connection to uaans69 closed.

[sysadmin@ansible-server ~]$ ansible uaans69 -m raw -a "who -r" -o

uaans69 | SUCCESS | rc=0 | (stdout) run-level 3 2018-05-14 13:19\r\n (stderr) Shared connection to uaans69 closed.\r\n

[sysadmin@ansible-server ~]$

“raw” module just executes a low-down and dirty SSH command over the network.

Shell module is very useful when you want to use redirections and shell’s inbuilt functionality.

1. Find out the disk utilization using “command” module.

[sysadmin@ansible-server ~]$ ansible -a "df -h" uaans69

uaans69 | SUCCESS | rc=0 >>

Filesystem Size Used Avail Use% Mounted on

/dev/sda2 18G 2.7G 14G 16% /

tmpfs 931M 0 931M 0% /dev/shm

/dev/sda1 283M 40M 229M 15% /boot

/dev/sr0 3.7G 3.7G 0 100% /mnt

[sysadmin@ansible-server ~]$

2. Can you try to filter the utilization only for “/” or “/dev/sda2”?

[sysadmin@ansible-server ~]$ ansible -a "df -h|grep sda2" uaans69

uaans69 | FAILED | rc=1 >>

df: invalid option -- '|'

Try `df --help' for more information.non-zero return code

[sysadmin@ansible-server ~]$

The above errors show that you can’t use the pipe (|) in the command module. Let’s use the “Shell” module.

3. Run the same command using the shell module.

[sysadmin@ansible-server ~]$ ansible -m shell -a "df -h|grep sda2" uaans69

uaans69 | SUCCESS | rc=0 >>

/dev/sda2 18G 2.7G 14G 16% /

[sysadmin@ansible-server ~]$ ansible -m shell -a "df -h|grep sda2" uaans69 -o

uaans69 | SUCCESS | rc=0 | (stdout) /dev/sda2 18G 2.7G 14G 16% /

[sysadmin@ansible-server ~]$

Here, we have got the desired results since “|” is shell’s inbuilt functionality.

4. Let’s test the redirection functionality on the command module.

[sysadmin@ansible-server ~]$ ansible -a "df -h > /var/tmp/df.out" uaans

uaans | FAILED | rc=1 >>

df: ‘>’: No such file or directory

df: ‘/var/tmp/df.out’: No such file or directorynon-zero return code

[sysadmin@ansible-server ~]$

Redirection failed when you use “command” module.  
Let’s run the same command using the shell module.

[sysadmin@ansible-server ~]$ ansible -m shell -a "df -h > /var/tmp/df.out" uaans

uaans | SUCCESS | rc=0 >>

[sysadmin@ansible-server ~]$

Verify the redirection by logging in to the remote server,

[sysadmin@ansible-server ~]$ ssh uaans

Last login: Fri Jul 6 23:33:26 2018 from 192.168.3.151

[sysadmin@uaans ~]$ ls -lrt /var/tmp/df.out

-rw-rw-r--. 1 sysadmin sysadmin 319 Jul 6 23:33 /var/tmp/df.out

[sysadmin@uaans ~]$ cat /var/tmp/df.out

Filesystem Size Used Avail Use% Mounted on

/dev/sda3 18G 4.9G 13G 28% /

devtmpfs 908M 0 908M 0% /dev

tmpfs 914M 0 914M 0% /dev/shm

tmpfs 914M 8.6M 905M 1% /run

tmpfs 914M 0 914M 0% /sys/fs/cgroup

/dev/sda1 297M 85M 213M 29% /boot

[sysadmin@uaans ~]$

5. Shell module offers to run a specific “shell” on the remote server as well.

[sysadmin@ansible-server ~]$ ansible -m shell -a "/bin/bash |echo $SHELL" uaans

uaans | SUCCESS | rc=0 >>

/bin/bash

[sysadmin@ansible-server ~]$

[sysadmin@ansible-server ~]$ ansible -m shell -a "/bin/bash |df -h " uaans

uaans | SUCCESS | rc=0 >>

Filesystem Size Used Avail Use% Mounted on

/dev/sda3 18G 4.9G 13G 28% /

devtmpfs 908M 0 908M 0% /dev

tmpfs 914M 0 914M 0% /dev/shm

tmpfs 914M 8.6M 905M 1% /run

tmpfs 914M 0 914M 0% /sys/fs/cgroup

/dev/sda1 297M 85M 213M 29% /boot

[sysadmin@ansible-server ~]$